Unix and Beyond:
Themes in Operating Systems Research

Dennis M. Ritchie
Bell Laboratories
Alcatel Lucent

dmr@bell-labs.com
http://www.cs.bell-labs.com/~dmr

Unix Origins (1969 - ...)

- Bell Labs has a long history in early operating systems, for example BE-SYS for IBM 709x machines; Multics
- Ken Thompson wanted to write a computer operating system by the 1960s:
  - Explore structures for building OSs
  - Build something for our own group to use
- Fundamental idea: a good way to represent data (disk files)
- First steps for Unix:
  - find concrete representation for data on disk
  - define access methods to data
- Earliest Unix simply tried to build some superstructure to test Thompson's ideas

Main Early Ideas (1969-72)

- Much was inherited, especially from Multics project
- Files contain just a sequence of bytes--
  - interpretation is up to applications
  - optimization of access is up to operating system
  - preference is for files with readable text, not binary
- Files are named in a hierarchical, tree-like name space, e.g.

```
/usr/dmr/japan/japanslides.ppt
```


File System access operations

- Basic operations are very simple:
  - `handle = open(filename, read-or-write-mode)`
  - `handle = create (filename, protection-mode)`
  - `read(handle, buffer, nbytes)`
  - `write(handle, buffer, nbytes)`
  - `close(handle)`
  - `seek(handle, place)`

Hierarchical Names

```
/
 -- source
   -- shell
         sh1.c
         sh2.c
         ...
   -- usr
      -- ken
         ...
      -- dmr
          -- japan
               japanslides.ppt
               ...
       ...
   -- bin
       sh
       ...
   -- ...
```

- These are standard now!

Names for other things

- Hierarchical naming is standard now
- Unix introduced a consistent abstraction: names for I/O devices as part of the same hierarchy, for example

  `/dev/tty23`
  (name of terminal 23)
  `/dev/disk/disk03`
  (name of a whole disk)
  `/dev/mem`
  (name for main memory)

- These really exist in the file system and the same protection and ownership properties are applied as with ordinary files
- To the extent possible, they look like the same byte streams as plain disk files
- Some (like terminals and networks) have some special operations that apply: “I/O controls”

Remote file systems

- By 1980s, Sun (with NFS), AT&T Computer Systems (RFS), and Bell Labs Research (NetA) were building "remote file systems" -- others pioneered also, like Xerox
  - Using RPC (remote procedure call) mechanisms, attach file system hierarchies on other machines to a local machine
- Important generalization: approach to a distributed system transparent to applications
- Some problems:
  - How usable are remote I/O devices?
    - *Some problems with protocols and blocking devices*
  - How usable are remote resources?
    - *Interworking across machine architectures, e.g. Differences in byte ordering*

Unix results and impact

- Influence has been substantial and influential:
  - Academic, commercial, government research use in 1970s, building on growth of minicomputer industry
  - Widespread use in 1980s following growth of workstation industry
  - Current industry and private interest centers on “open software” movement, enabled by open Unix standards
  - Other operating systems have been much influenced by these standards

Another step: Plan 9 (1990-2003)

- Original idea: define a remote protocol (called 9P) to talk to the file system
- Unix began with a structural idea:
  How to represent a local file system?
- Plan 9 began with this structural idea:
  How to talk to resources (local or remote)?

Plan 9 idea #1

- Make all resources accessible by file system names:

```
/dev/console
/dev/consctl
/dev/mouse
/proc    (information about processes)
  mem
    stack
    status
    control
    ...
/net
   dns    (domain name service)
   tcp
      clone   (create connections)
      tcp05
          data  (send/receive data)
          control
          status
        tcp10
    ....
```


Plan 9 Idea #2

- The naming scheme is private to each group of processes
  - for example, for those of a user of a shared machine
  - or for programs running in a window of a window system
- Each user's programs can locally readjust their name space
  - by *binding* existing parts of the current name space of files
  - by *mounting* resources supplied by a server of the 9P protocol

Example: Network access

- Access to network devices is through files
- Pieces of another machine's file space can be imported
- Thus, programs on one machine can borrow network interfaces from servers
- For example, on a machine on a Bell Labs internal network:

```
% telnet stanford.edu
telnet: Cannot translate address stanford.edu
%
```


Example: Network access

- Access to network devices is through files
- Pieces of another machine's file space can be imported
- Thus, programs on one machine can borrow network interfaces from servers
- For example, on a machine on a Bell Labs internal network:

% telnet stanford.edu

```
telnet: Cannot translate address stanford.edu
```

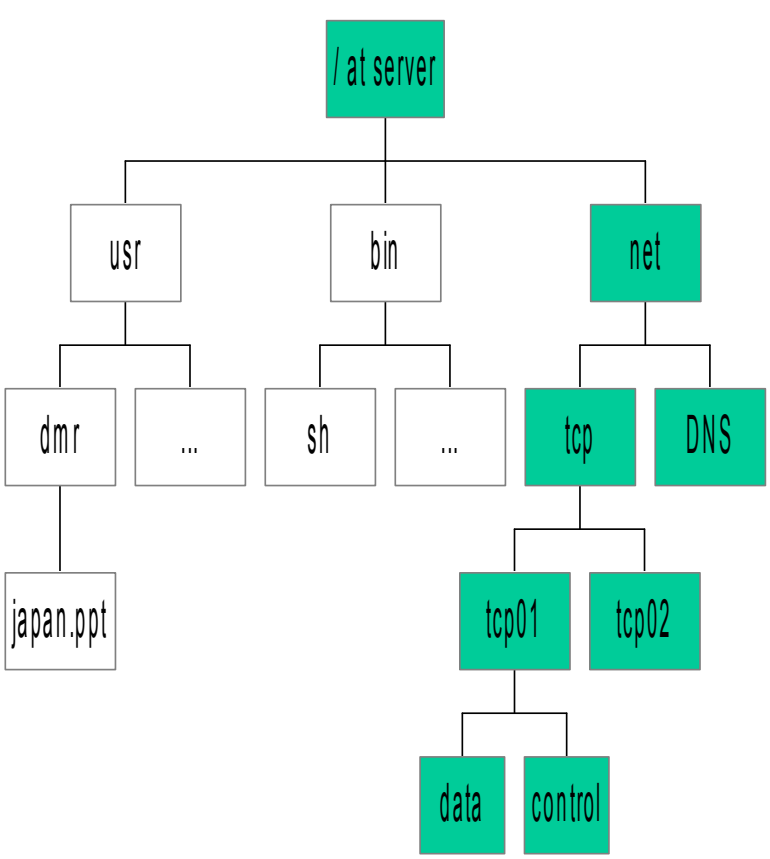
% import internetgate.bell-labs.com /net

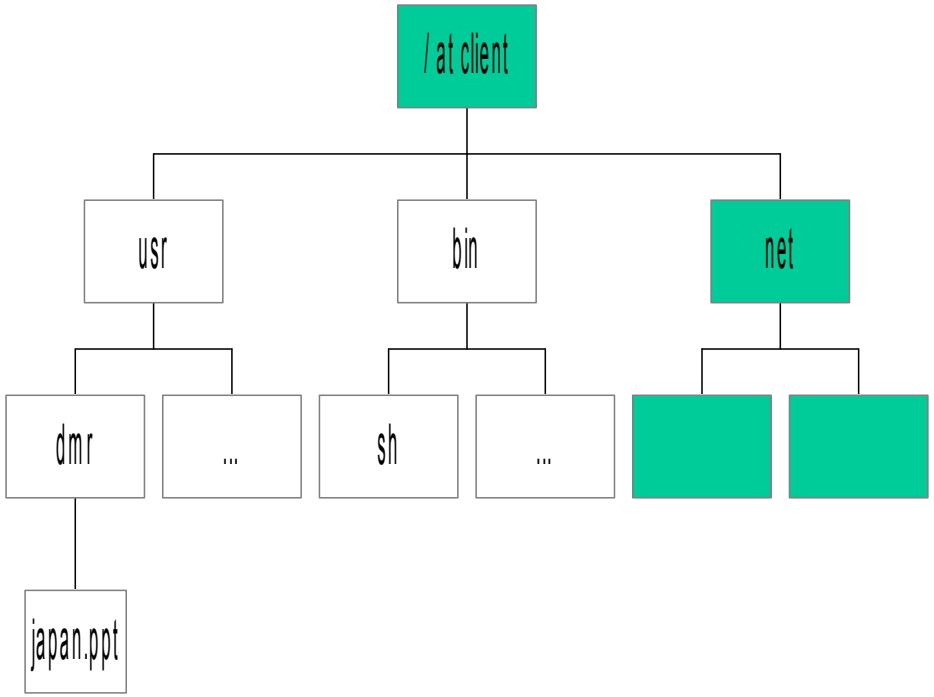
(this binds the Internet gateway's network directory onto my own; now my /net is the one on the server machine, connected to the Internet)

% telnet stanford.edu

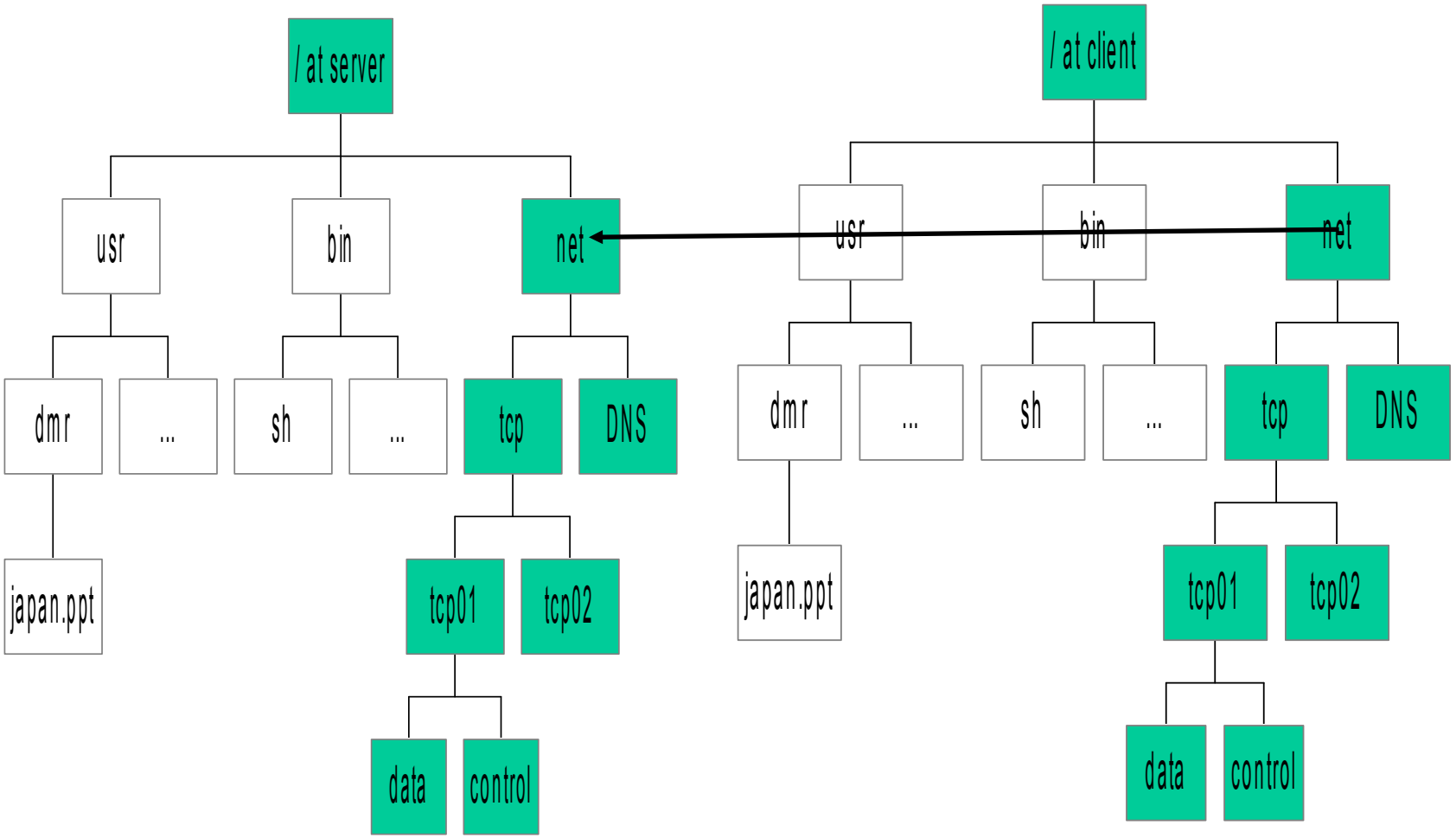
```
connected to tcp!stanford.edu!telnet on /net/tcp/15
 UNIX(r) System V Release 4.0 (norob) (pts/1)
login:
```


Network Example (before binding)





Network Example (after binding)



Plan 9 Example 2

- Exotic file systems can be created, building on standard, old protocols like “File Transfer Protocol”-- FTP
- Works by changing name space so that files in
  `/n/ftp`
  actually access a server that turns reads and writes into FTP requests:

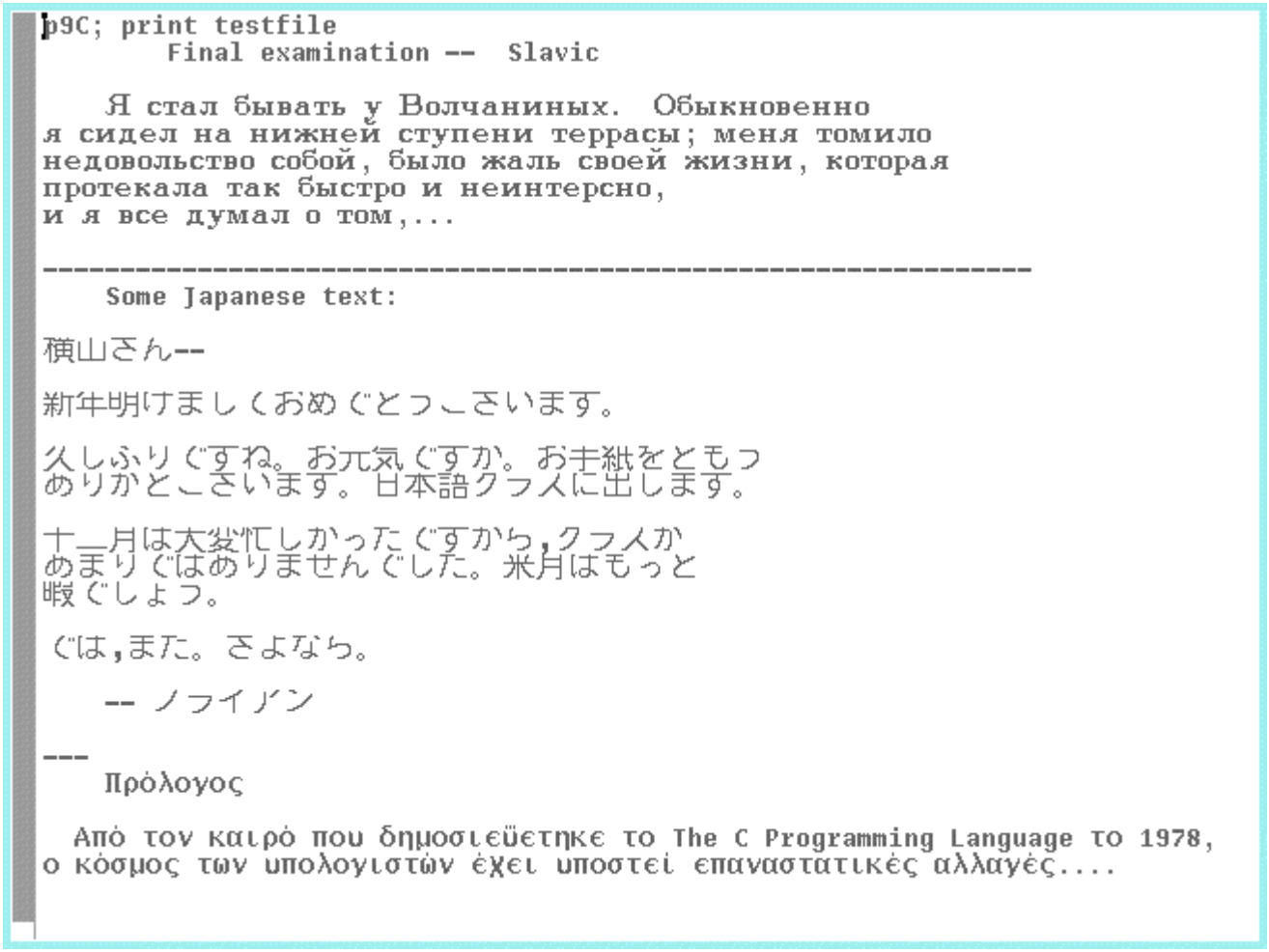
```
% ftpfs ftp.microsoft.com /n/ntp
230-This is FTP.MICROSOFT.COM
230 Anonymous user logged in.
215 Windows_NT version 5.0
% ls -l /n/ftp
dr-xr-xr-x owner           0  Apr 18 14:10  KBHelp
-r-xr-xr-x owner       54310  Apr 19 03:52  LS-LR.ZIP
-r-xr-xr-x owner       28160  Apr 19 03:52  MSNBRO.DOC
  ...
```


Plan 9 Idea #3

- Try to make data machine-independent
  - use text representation where possible: encourage text instead of binary messages when possible
  - use files to connect programs, for example, write
    `connect tcp!135.104.3.11!http`
    on a control file instead of transmitting a binary structure
  - Text works across remote file systems, independent of byte ordering
- Encourage a more universal character set, like Unicode (ISO 10646) in UTF-8 encoding: helpful for internationalization--even perhaps for 日本語

Screen-shot example

A Plan 9 image:



Recent Lucent Results/Products/Spinoffs

- The Plan 9 distribution

  `http://plan9.bell.labs.com/plan9`

  is an open distribution of the Plan 9 research work

- Inferno ™
- Lucent Products (& former products)
  - Pathstar™ Access Server
  - Lucent Managed Firewall™
- Not yet products
  - Viaduct

Inferno System

- An OS that combines the system structure ideas from Plan 9 with other ideas:
  - A *virtual operating system* that can run either stand-alone on a small device
    - hand-held, or set-top box, games console
  - Or as an ordinary application under Windows, Unix, etc.
- By chance and circumstance, similar for portable languages and systems were also re-emerging with Java language technology
- Lucent's Inferno business was transferred to Vita Nuova:

`http://www.vitanuova.com`

Pathstar Access Server

- An Internet Protocol router designed to be useful for telephony applications as an end-office (Class 5) telephone switch; it handled
- Ordinary voice telephone with usual features
  - Data applications (IP using standard protocols)
  - A wide variety of end-connection mechanisms
    (plain telephone, fiber links, etc.)
  - Used Inferno for its operating system

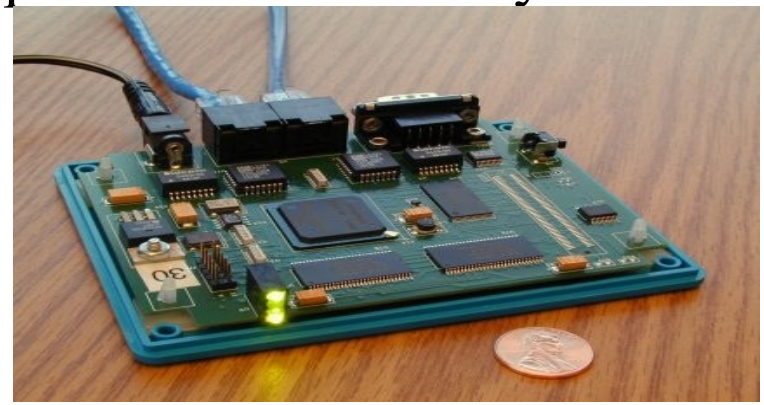
Lucent Managed Firewalls

- An Internet-to-Intranet safety interface, with packet and content filtering
- Uses Inferno as its internal operating system for its central element: the “brick”.

`http://www.lucent.com/ins/products/vpnfirewall/index.html`

Viaduct

- A small box (15 cm long) provides VPN (Virtual Private Network) secure tunneling for homes or small offices
- Does encryption and compression
- Intended for DSL and cable modem connections
  - no administration needed--just insert between modem and computer
- Uses Plan 9 as its operating system
- Not a product: used mainly in research group



FeaVer

- Project for showing correctness of protocols and software
- Uses Holzmann's **spin** model-checking technology
- Used in Pathstar for checking call-processing code
- A multi-CPU system was built as a tool for this; it runs Plan 9

Summary

- The line of research has been highly fruitful
  - Unix and its offspring have been successful and influential (especially lively today in Linux)
  - Adoption of our technology is pervasive throughout the computer world
- It still continues: but the challenge for Bell Labs research is to continue to find the new and interesting places in which to work
- E.g. Factotum / Secure Store, Venti (Presotto, Grosse, Quinlan);

  UMTS/HSDPA high-speed wireless (Bosch/Mullender)

Unix and Beyond:
Themes in Operating Systems Research

Dennis M. Ritchie
Bell Laboratories
Alcatel Lucent

dmr@bell-labs.com
http://www.cs.bell-labs.com/~dmr